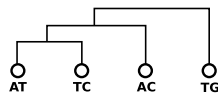


6.095/6.895 Fall 2005 Practice Midterm

These problems are representative of the material for the midterm but may not approximate it in either length or difficulty. They are meant to help you review material. The midterm will have a combination of quick-to-answer knowledge questions, short practical exercises along with more involved problems that require a thorough understanding of the material. No books, notes or electronic aids (such as calculators) will be permitted.

1. Name two clustering algorithms (or paradigms) and explain two applications to biological problems.
2. What do DNA microarrays measure?
3. Give and contrast algorithms for motif finding. When would one be most useful? Analyze their running time and space requirements.
4. Dynamic programming is used pervasively in computational biology. Give two problems in computational biology that are solved with dynamic programming and explain what biological problems they solve.
5. When is a distance matrix additive? When is it ultrametric?
6. What are some of the weaknesses of UPGMA and how are they addressed by neighbor-joining?
7. What does the Viterbi algorithm solve? What class of algorithms does it belong to?
8. What are the biological ideas behind a sequence motif? What are ways that we can represent these motifs?
9. How would you generate a sample sequence from an HMM model? Given an HMM, how do you compute $Pr(x|M)$?
10. Give two probability problems that can be solved with the forward and backwards algorithms.
11. In lecture, we learned about 4 typical problems related to HMMs. Indicate the probability problems these solve.
12. Gibbs sampling is an iterative algorithm, what does it do at each iteration? In motif discovery after a profile is created and the positions in the chosen DNA sequence scored, how is the new starting position in this DNA sequence chosen?
13. How does Gibbs sampling relate to EM? How does EM differ from a purely greedy approach to determining the next iterations profile?
14. How often are perfectly conserved k -mers expected in random sequences? How often do they appear in biological sequences (more or less, and why)? How did we cope with the absence of contiguous k -mers (degenerate motifs)?
15. What is the most parsimonious explanation for the following tree (assume a uniform cost function):



16. How do we find the connected components of a graph?
17. What problem does bisection solve and how do we approximate its solution?
18. In what circumstances is dimensionality reduction useful? How can we perform dimensionality reduction using SVD?
19. What is the purpose of pseudo-counts? Give two examples of where we have used them in class.
20. How does the BLAST algorithm work? What are the benefits/drawbacks of BLAST over performing individual alignments?
21. What are suffix trees and why would we create them?

22. Compare and contrast local and global alignments. What are dynamic programming update rules for each?
23. Give an Karp-Rabin style update rule for the hash function $h(s) = \sum_i s_i \pmod{7}$ where s_i denotes the i th digit of s .
24. What are two enumeration strategies for finding a common string between several sequences, with at most 2 mismatches.
25. Given a motif

	1	2	3	4
A	0.0	0.5	0.0	0.0
T	0.0	0.5	0.0	1.0
G	1.0	0.0	0.5	0.0
C	0.0	0.0	0.5	0.0

and single sequence

$$s = \text{ACGAGTGTCTA}$$

give the motif found at the next iteration of the EM algorithm when you have pseudo-counts of 1.

26. Number the nodes for the graph represented by the adjacent matrix below as they would be by the connected component algorithm given in class:

	a	b	c	d	e	f	g	h	i
a	0	1	0	0	1	0	0	0	0
b	1	0	0	0	0	0	1	0	0
c	0	0	0	1	0	0	0	0	1
d	0	0	1	0	0	0	0	0	1
e	1	0	0	0	0	0	1	1	0
f	0	0	0	0	0	0	0	0	0
g	0	1	0	0	1	0	0	1	0
h	0	0	0	0	1	0	1	0	0
i	0	0	1	1	0	0	0	0	0

27. Give an $O(nm)$ algorithm to find an optimal global alignment when there is a affine *mismatch* penalty.
28. Give an estimate (either an upper, lower or exact bound) for the number of possible local alignments for a sequence of length n and one of length m .
29. How many internal nodes does a hierarchical tree on n elements have? Hint: consider the algorithm for constructing this tree.
30. Give a good upper bound on the number of hierarchical clusterings on n elements.
31. Show that the location of the root does not affect the substitution cost when performing parsimony.